

Docket No. AUS000181US1

CLAIMS:

What is claimed is:

- 5 1. An application development server, comprising:
a storage unit for storing a plurality of working
definitions for a plurality of computing environments;
and
an interface for receiving requests for one of the
10 plurality of computing environments from and transferring
the requested one of the plurality of computing
environments to a client over a communications line;
wherein
the working definition for each computing
15 environment is sufficient from which to construct a valid
runtime image of the respective computing environment.
2. The application development server as recited in
claim 1, wherein the network is the Internet.
- 20 3. The application development server as recited in
claim 1, the working definitions define roles for
application development and support the roles.
- 25 4. The application development server as recited in
claim 1, wherein the working definitions define
application development team members and a role each team
member holds.
- 30 5. The application development server as recited in
claim 1, wherein the encapsulated development environment

Docket No. AUS000181US1

comprises source code for the application.

6. The application development server as recited in claim 1, wherein encrypted links are sent to the client
5 along with the encapsulated development environment, the encrypted links providing support for development, testing, beta testing, and deployment of the application.

7. The application development server as recited in
10 claim 1, wherein the requested one of the plurality of computing environments supports the requesting clients role in a parallel processing application.

8. The application development server as recited in
15 claim 1, wherein the working definition supports management of a workflow application.

9. The application development server as recited in
20 claim 1, wherein the network is an intranet.

10. A method of providing an encapsulated development environment for automated software development to a client, the method comprising:

receiving a request for the encapsulated development
25 environment, wherein the encapsulated development environment comprises platform independent definitions defining an application and environment from which a runtime representation of the environment may be constructed;

30 retrieving the encapsulated development environment;
and

Docket No. AUS000181US1

sending the encapsulated development environment to the client.

11. The method as recited in claim 10, further comprising:

receiving an update to the encapsulated development environment; and

modifying the encapsulated development environment to reflect such update.

10

12. A method of providing different views into a construct, the method comprising:

detecting a change within a data processing system;

responsive to a determination that the change

effects a working definition of the construct, modifying the working definition of the construct to reflect a change indicated by the change; and

updating the runtime representations of the construct.

20

13. The method as recited in claim 12, wherein the change is an event.

14. A computer program product in computer readable media for use in a data processing system for providing an encapsulated development environment for automated software development to a client, the computer program product comprising:

first instructions for receiving a request for the encapsulated development environment, wherein the encapsulated development environment comprises platform

Docket No. AUS000181US1

independent definitions defining an application and environment from which a runtime representation of the environment may be constructed;

second instructions for retrieving the encapsulated development environment; and

third instructions for sending the encapsulated development environment to the client.

15. The computer program product as recited in claim 14, further comprising:

fourth instructions for receiving an update to the encapsulated development environment; and

fifth instructions for modifying the encapsulated development environment to reflect such update.

15

16. A computer program product in computer readable media for use in a data processing system for providing different views into a construct, the method comprising:

first instructions for detecting a change within a data processing system;

second instructions, responsive to a determination that the change effects a working definition of the construct, for modifying the working definition of the construct to reflect a change indicated by the change;

25 and

third instructions for updating the runtime representations of the construct.

17. The computer program product as recited in claim 16, wherein the change is an event.

30

Docket No. AUS000181US1

18. A system for providing an encapsulated development environment for automated software development to a client, comprising:

5 means for receiving a request for the encapsulated development environment, wherein the encapsulated development environment comprises platform independent definitions defining an application and environment from which a runtime representation of the environment may be constructed;

10 means for retrieving the encapsulated development environment; and

means for sending the encapsulated development environment to the client.

15 19. The system as recited in claim 18, further comprising:

means for receiving an update to the encapsulated development environment; and

20 means for modifying the encapsulated development environment to reflect such update.

20. A system for providing different views into a construct, comprising:

25 means for detecting a change within a data processing system;

means, responsive to a determination that the change effects a working definition of the construct, for modifying the working definition of the construct to reflect a change indicated by the change; and

30 means for updating the runtime representations of the construct.

Docket No. AUS000181US1

21. The system as recited in claim 20, wherein the change is an event.

5 22. A memory for storing data for access by an application program executed on a data processing system, comprising:

a data structure stored in the memory, the data structure comprising:

10 a plurality of conditions; and

a plurality of working definitions; wherein

each of the plurality of working definitions is sufficient from which to construct a valid runtime representation of a corresponding application; and

15 the plurality of conditions determine which of the plurality of working definitions are expressed on the occurrence of an event.

20 23. The memory as recited in claim 22, wherein the plurality of conditions determine which of the plurality of working definitions are suppressed on the occurrence of an event.

25 24. A synchronization facility for synchronizing various views of a construct: comprising:

detection means for detecting changes to the computer system;

comparison means for comparing the changes to working definitions of applications;

30 updating means for updating working definitions of effected applications to reflect the changes thereby

Docket No. AUS000181US1

producing updated working definitions; and

construction means for constructing an updated valid runtime representation of the effected applications using updated working definitions.

5